User Model User-Adap Inter (2018) 28:97–125
https://doi.org/10.1007/s11257-018-9202-0

Coherence and inconsistencies in rating behavior: estimating the magic barrier of recommender systems

Alan Said[1] · Alejandro Bellogín[2]

Received: 10 May 2017 / Accepted in revised form: 2 April 2018 / Published online: 13 April 2018

Abstract Recommender Systems have to deal with a wide variety of users and user types that express their preferences in different ways. This difference in user behavior can have a profound impact on the performance of the recommender system. Users receive better (or worse) recommendations depending on the quantity and the quality of the information the system knows about them. Specifically, the inconsistencies in users' preferences impose a lower bound on the error the system may achieve when predicting ratings for one particular user—this is referred to as the *magic barrier*. In this work, we present a mathematical characterization of the magic barrier based on the assumption that user ratings are afflicted with inconsistencies—noise. Furthermore, we propose a measure of the consistency of user ratings (rating coherence) that predicts the performance of recommendation methods. More specifically, we show that user coherence is correlated with the magic barrier; we exploit this correlation to discriminate between *easy* users (those with a lower magic barrier) and *difficult* ones (those with a higher magic barrier). We report experiments where the recommendation error for the more coherent users is lower than that of the less coherent ones. We further validate these results by using two public datasets, where the necessary data to identify the magic barrier is not available, in which we obtain similar performance improvements.

Alan Said and Alejandro Bellogín have contributed equally to this work.

✉ Alan Said
alansaid@acm.org

✉ Alejandro Bellogín
alejandro.bellogin@uam.es

[1] University of Skövde, Skövde, Sweden

[2] Universidad Autónoma de Madrid, Madrid, Spain

Keywords Recommender systems · Evaluation · Benchmarking · Noise · Rating coherence · Context · Evaluation metrics · Magic barrier · Ratings · User behavior

1 Introduction

Recommender systems play an important role in most top-ranked commercial websites such as Amazon, Netflix, Last.fm or IMDb (Ricci et al. 2011). The goal of these recommender systems is to increase revenue and present personalized user experiences by providing suggestions for previously unknown items that are potentially interesting for a user. With the growing amount of information available on the Internet, the importance of good recommender systems increases even more as a means to guide users through the massive amounts of available data.

The key role of recommender systems has resulted in a vast amount of research in this field, which has yielded a plethora of different recommender algorithms (Desrosiers and Karypis 2011; Ricci et al. 2011; Adomavicius and Tuzhilin 2005; Lathia et al. 2010). An example of a popular and widely used approach to recommendation is *collaborative filtering*. Collaborative filtering computes user-specific recommendations based on historical user data, such as ratings or other usage patterns (Desrosiers and Karypis 2011; Koren and Bell 2011). Other approaches include content-based recommenders (suggesting items based on properties and content of a specific item), social recommenders (suggesting items based on past behavior of similar users in a social graph) or hybrid combinations of several different approaches.

To select an appropriate recommender algorithm and adapt it to a given scenario or problem, the algorithms are usually examined by testing their performance using either artificial or real test data reflecting the problem. The best performing algorithm and parameters among a number of candidate algorithms is chosen. To be able to compare performance, several measures and metrics have been defined. Common measures are precision and recall, normalized discounted cumulative gain (NDCG), receiver operating characteristic (ROC) or the root-mean-squared error (RMSE). RMSE has perhaps been the most popular metric used to evaluate the prediction accuracy of a recommender algorithm (Shani and Gunawardana 2011); as a matter of fact, it was the central evaluation metric used in the Netflix Prize.[1] However, in recent research RMSE is being phased out and replaced by ranking-based metrics (like NDCG or precision), in part due to its inability of discriminating the error produced by the recommender between items with higher predicted values (which would appear at the top if a ranking has to be produced) and items with lower predicted values (McLaughlin and Herlocker 2004; McNee et al. 2006). Moreover, other methods to rank potential items of interest to the user can be exploited besides the highest predicted rating, such as majority vote or others (Masthoff 2015), which RMSE and related error-based metrics cannot measure properly.

Nonetheless, it is important to understand what happens when the error in rating prediction is being optimized, or, in other terms, when RMSE is used as a performance measure in evaluation. In this situation, the recommendation task is typically posed

[1] http://www.netflixprize.com.

as that of learning a rating function that minimizes the RMSE on a given training set of user ratings. The generalized RMSE performance of the learned rating function is then assessed on some independent test set of ratings, which is disjoint from the training set. One major drawback of measuring and comparing the performance using only static, previously collected, test data is that user behavior in the data is not always reliable. According to previously published studies, e.g., (Amatriain et al. 2009a; Hill et al. 1995), user ratings can be inconsistent (noisy) in the sense that a user may rate the same item differently at different points in time. Following these findings, Herlocker et al. (2004) coined the term *the magic barrier*. The magic barrier marks the point at which the performance and accuracy of a recommendation algorithm cannot be enhanced due to inherent noise in the data. Every improvement in accuracy might denote overfitting on noise and other inconsistencies, and not improved actual performance. Thus, comparing and measuring the expected future performance of algorithms based on static data may not work.

We propose a method to infer which users have a higher level of inconsistency a priori, that is, without any extra information or by studying the quality of the recommendations received. In particular, we are interested in relating this to the magic barrier, in such a way that one could predict which users will have a low or high magic barrier. We present a measure of user coherence which takes into consideration how the user assigns ratings within an item's feature space. By doing so, we associate highly coherent values to users with a lower magic barrier, that is, those having a more consistent rating behavior.

Once the magic barrier—or any other measure of users' inconsistency—is successfully predicted, it is possible to improve the recommender system's performance—either the accuracy or precision of the recommendations, but also the computational effort of the adopted techniques—by exploiting the user coherence, as long as rating information is available in the system. Although implicit feedback datasets and systems where no ratings are provided by the user are growing increasingly popular, there is still a substantial amount of literature based on rating-based recommender systems, and several popular sites (e.g., Amazon[2], IMDb[3], TripAdvisor[4], Yelp[5]) still use them today. Hence, we believe this research is relevant and could help improve the performance of the algorithms being used in those, and in other, situations.

Our research aims to answer the following two research questions:

RQ1 how good of a predictor for the magic barrier is the rating coherence of a user? and

RQ2 is it possible to cluster the user community into easy and difficult users—according to their coherence—so that the performance of the system is improved, and if so, by how much?

We address the first question by measuring the correlation between our definition of coherence and the magic barrier of each user. For the second question, we create

[2] https://www.amazon.com.

[3] http://www.imdb.com.

[4] https://www.tripadvisor.com.

[5] http://www.yelp.com.

separate training models for a subset of the users according to their predicted consistency: the user community is clustered into *easy* and *difficult* users—according to their coherence; these groups allow us to study how the error of the recommender system changes depending on the observed rating inconsistency. We would like to emphasize that, throughout our study, there is only one single dataset (or system) for which the magic barrier data is available. This is mostly because it is a difficult experiment to run that requires multiple ratings from the same items over time. Thus, the correlation validation is limited to this single instance; however, we develop a general framework for the second question that can be applied to datasets where the necessary data to identify the magic barrier is not available, allowing us to exploit and validate the proposed rating coherence in more domains and situations.

Our research, hence, provides a measure of the noise present in user behavior, inferred from the ratings available in a recommender system. This measure (user coherence) is shown to successfully predict the magic barrier of the system, and when used to classify the users according to their performance, the performance of the complete system is improved. We present in this paper different instantiations of the user coherence, all of them using information readily available in the recommender system; we also show—from an analytical point of view—under which conditions it would be equivalent to the magic barrier.

Therefore, our main contributions include:

- A derivation of the so-called magic barrier from a statistical learning theory point of view by posing the recommendation task as a risk minimization problem, summarizing the main findings presented in (Said et al. 2012a).
- A general model for user coherence that fits into the magic barrier formulation and can be computed using only information available at training time by the recommendation algorithm, extending the formulation included in (Bellogín et al. 2014).
- We use this model in three datasets with different characteristics and show how the overall error of the system is affected by the number of coherent users involved.

The remainder of this paper is structured as follows: in Sect. 2 we present the magic barrier as it will be used throughout the paper, Sect. 3 presents our approaches to measure the coherence of a user, Sect. 4 validates the proposed measure as an estimator of the magic barrier by using a correlation analysis, Sect. 5 presents the datasets, experimental settings, and results obtained, and Sect. 6 provides additional works dealing with the problem of predicting the user's difficulty or the performance of a system; finally, in Sect. 7 we conclude the paper and present some lines of future work.

2 A statistical model for users' inconsistencies

2.1 Recommendation as risk minimization

We pose the recommendation task as that of a function regression problem based on the empirical risk minimization principle from statistical learning theory (Vapnik 1995). This setting provides the theoretical foundation to derive a lower bound on the root-mean-square error (henceforth referred to as the magic barrier) that can be attained by an optimal recommender system.

We begin by describing the traditional setting of a recommendation task as presented in (Desrosiers and Karypis 2011). Suppose that $\mathcal{R}$ is a set of ratings r_{ui} submitted by users $u \in \mathcal{U}$ for items $i \in \mathcal{I}$. Ratings may take values from some discrete set $\mathcal{S} \subseteq \mathbb{R}$ of rating scores. Typically, ratings are known only for few user-item pairs. The recommendation task consists of suggesting new items that will be rated highly by users.

It is common practice to pose the recommendation task as that of learning a rating function

$$f : \mathcal{U} \times \mathcal{I} \to \mathcal{S}, \quad (u, i) \mapsto f(u, i)$$

on the basis of a set of training examples from $\mathcal{R}$. Given a user u, the learned rating function f is then used to recommend those items i that have the largest scores $f(u, i)$. The accuracy of a rating function f is evaluated on a test set, which is a subset of $\mathcal{R}$ disjoint from the training set.

A popular and widely used measure for evaluating the accuracy of f on a set $\mathcal{R}$ of ratings is the root-mean-square error criterion

$$E(f|\mathcal{R}) = \sqrt{\frac{1}{|\mathcal{R}|} \sum_{(u,i)\in\mathcal{R}} \big(f(u, i) - r_{ui}\big)^2}, \tag{1}$$

where the sum runs over all user-item pairs (u, i) for which $r_{ui} \in \mathcal{R}$.[6]

Learning a rating function by minimizing the RMSE criterion can be justified by the inductive principle of empirical risk minimization from statistical learning theory (Vapnik 1995). Within this setting we describe the problem of learning a rating function as follows: we assume that

- user-item pairs (u, i) are drawn from an unknown probability distribution $p(u, i)$,
- rating scores $r \in \mathcal{S}$ are provided for each user-item pair (u, i) according to an unknown conditional probability distribution $p(r|u, i)$,
- $\mathcal{F}$ is a class of rating functions.

The probability $p(u, i)$ describes how likely it is that user u rates item i. The conditional probability $p(r|u, i)$ describes the probability that a given user u rates a given item i with rating score r. The class $\mathcal{F}$ of functions describes the set from which we choose (learn) our rating function f for recommending items. An example of $\mathcal{F}$ is the class of nearest neighbor-based methods (Desrosiers and Karypis 2011).

The goal of learning a rating function is to find a function $f \in \mathcal{F}$ that minimizes the expected risk function

$$R(f) = \sum_{(u,i,r)} p(u, i, r)\big(f(u, i) - r\big)^2, \tag{2}$$

[6] For the sake of brevity, we abuse notation and write $(u, i) \in \mathcal{R}$ for user-item pairs (u, i) for which $r_{ui} \in \mathcal{R}$.

where the sum runs over all possible triples $(u, i, r) \in \mathcal{U} \times \mathcal{I} \times \mathcal{S}$ and $p(u, i, r) = p(u, i)p(r|u, i)$ is the joint probability.

The problem of learning an optimal rating function is that the distribution $p(u, i, r)$ is unknown. Therefore, we cannot compute the optimal rating function

$$f_* = \arg\min_{f \in \mathcal{F}} R(f)$$

directly. Instead, we approximate f_* by minimizing the empirical risk

$$\widehat{R}(f|\mathcal{X}) = \frac{1}{|\mathcal{X}|} \sum_{r_{ui} \in \mathcal{X}} \big(f(u, i) - r_{ui}\big)^2,$$

where $\mathcal{X} \subseteq \mathcal{R}$ is a training set consisting of ratings r_{ui} given by user u for item i. Observe that minimizing the empirical risk is equivalent to minimizing the RMSE criterion.

A theoretical justification of minimizing the RMSE criterion (or the empirical risk) arises from the following result of statistical learning theory (Vapnik 1995): under the assumption that the user ratings from $\mathcal{R}$ are independent and identically distributed, the empirical risk is an unbiased estimate of the expected risk.[7]

2.2 Deriving the magic barrier

It is possible to derive a lower bound on the RMSE that can be attained by an optimal recommender system: the so-called magic barrier. We show that this magic barrier is the standard deviation of the inconsistencies (noise) inherent in user ratings. To this end, we first present a noise model and then derive the magic barrier.

As shown in previous user studies (Amatriain et al. 2009a; Hill et al. 1995), users' ratings tend to be inconsistent. Inconsistencies in the ratings could be due to, for example, change of taste over time, personal conditions, inconsistent rating strategies, and/or social influences, just to mention a few. For the sake of convenience, in this work we regard inconsistencies in user ratings as noise. The following fictitious scenario illustrates the basic idea behind our noise model: consider a movie recommender with n items and a rating scale from zero to five stars, where zero stars refers to a rating score reserved for previously un-rated items only. Users are regularly asked to rate m randomly selected items. After a sufficiently long period of time, each user has rated each item (movie) several times. The ratings may vary over time due to several reasons (see for instance Lathia et al. 2010) such as change of taste, current emotional state, group-dynamic effects, and other external as well as internal influences.

Keeping the above scenario in mind, the *expected rating* of a user $u \in \mathcal{U}$ on an item $i \in \mathcal{I}$ is defined by the expectation

$$\mathbb{E}[R_{ui}] = \mu_{ui},$$

[7] The set of users and items are both finite. In order to apply the law of large numbers, we may think of $\mathcal{R}$ as being a set of ratings obtained by randomly selecting triples (u, i, r) according to their joint distribution.

where R_{ui} is a random variable on the user-item pair (u, i) and takes values from some discrete set $\mathcal{S} \subseteq \mathbb{R}$ of rating scores. Then, a rating $r_{ui} \in \mathcal{R}$ is composed of the expected rating μ_{ui} and some error term ε_{ui} for the noise incurred by user u when rating item i. We occasionally refer to the error ε_{ui} as user-item noise. Thus, user ratings arise from a statistical model of the form

$$r_{ui} = \mu_{ui} + \varepsilon_{ui}, \tag{3}$$

where the random error ε_{ui} has expectation $\mathbb{E}[\varepsilon_{ui}] = 0$.

Suppose that f_* is the true, unknown, rating function that knows all expected ratings μ_{ui} of each user u on any item i, that is

$$f_*(u, i) = \mu_{ui} \tag{4}$$

for all users $u \in \mathcal{U}$ and items $i \in \mathcal{I}$. Then, the optimal rating function f_* minimizes the expected risk function Eq. (2). Substituting Eqs. (3) and (4) into the expected risk function Eq. (2) and using $p(u, i, r) = p(u, i)p(r|u, i)$ gives

$$R(f_*) = \sum_{(u,i)} p(u, i)\mathbb{E}\left[\varepsilon_{ui}^2\right] = \sum_{(u,i)} p(u, i)\mathbb{V}\left[\varepsilon_{ui}\right], \tag{5}$$

where the sum runs over all possible user-item pairs $(u, i) \in \mathcal{U} \times \mathcal{I}$ and $\mathbb{V}[\varepsilon_{ui}]$ denotes the variance of the user-item noise ε_{ui}. Equation (5) shows that the expected risk of an optimal rating function f_* is the mean variance of the user-item noise terms.

Expressed in terms of the RMSE criterion, the magic barrier $B_{\mathcal{U}\times\mathcal{I}}$ of a recommender system with users $\mathcal{U}$ and items $\mathcal{I}$ is then defined by

$$B_{\mathcal{U}\times\mathcal{I}} = \sqrt{\sum_{(u,i)} p(u, i)\mathbb{V}\left[\varepsilon_{ui}\right]}.$$

To put it in other terms: the magic barrier is the RMSE of an optimal rating function f_*. We note that even an optimal rating function has a non-zero RMSE *unless all users are continuously consistent in their ratings*; this value will be larger for any non-optimal function $f \in \mathcal{F}$. Observe that an optimal rating function needs not to be a member of our chosen function class $\mathcal{F}$ from which we select (learn) our actual rating function f. Thus the RMSE of f can be decomposed into the magic barrier $B_{\mathcal{U}\times\mathcal{I}}$ and an error E_f due to model complexity of f giving

$$E_{RMSE}(f) = B_{\mathcal{U}\times\mathcal{I}} + E_f > B_{\mathcal{U}\times\mathcal{I}}.$$

Finally, we estimate the magic barrier according to the procedure outlined in Algorithm 1, since, as in the case of the expected risk, we are usually unable to directly determine the magic barrier $B_{\mathcal{U}\times\mathcal{I}}$.

We postulate that all rating functions $f \in \mathcal{F}$ with an empirical risk of the form

$$\widehat{R}(f|\mathcal{X}) \leq \widehat{B}_{\mathcal{X}}^2$$

Algorithm 1 Procedure for estimating the magic barrier.

```
Procedure: Let $\mathcal{X} \subseteq \mathcal{U} \times \mathcal{I}$ be a randomly generated subset of user-item pairs.
1. For each user-item pair $(u, i) \in \mathcal{X}$ do
    (a) Sample $m$ ratings $r_{ui}^1, \ldots, r_{ui}^m$ on a regular basis
    (b) Estimate the expectation $\mu_{ui}$ by the sample mean
            $\widehat{\mu}_{ui} = \frac{1}{m} \sum_{t=1}^{m} r_{ui}^t$
    (c) Estimate the variance of the ratings
            $\widehat{\varepsilon}_{ui}^2 = \frac{1}{m} \sum_{t=1}^{m} \left(\widehat{\mu}_{ui} - r_{ui}^t\right)^2$
2. Estimate the magic barrier by taking the average
            $\widehat{B}_{\mathcal{X}} = \sqrt{\frac{1}{|\mathcal{X}|} \sum_{(u,i) \in \mathcal{X}} \widehat{\varepsilon}_{ui}^2}.$    (6)
```

are likely to overfit on the set $\mathcal{X}$ and consider any further improvements on the RMSE below $\widehat{B}_{\mathcal{X}}$ to be *meaningless*, because they might denote overfitting on noise, and not improving actual performance.

2.3 Capturing the magic barrier: an experiment with Moviepilot

Moviepilot[8] is a commercial movie recommender system which, at the time of the experiment, had more than one million users, 55, 000 movies, and over 10 million ratings. Movies are rated on a 0 to 10 scale with step size 0.5 (0 corresponding to a rating score of 0, not an unknown rating). This dataset contains several movie features such as emotion keywords, intended audience, time keywords, etc. As described in (Said et al. 2012a), to estimate the magic barrier a Web-based study was created for collecting users' *opinions* on movies. An opinion is a score in the same rating scale as standard user ratings, however they do not show up in users' profiles and are only stored in the survey, subsequently, not affecting the recommendations given to the users.

Whilst taking part in the study, users were presented with a number of movies randomly drawn from the complete set of their rated movies. Each user could submit at most one opinion on each movie. A user could skip any number of movies without providing any opinion. After at least 20 opinions had been given, the user could complete the study. The study ran from mid April 2011 to early May 2011, where only users who provided opinions on at least 20 movies were recorded. A total of 306 users provided 6299 opinions about 2329 movies.

[8] http://www.moviepilot.de/.

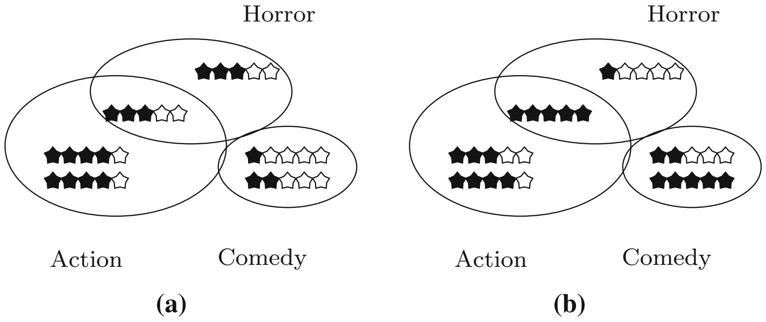

Fig. 1 Example of a coherent (u_1) versus not coherent (u_2) user, using Eq. 7 and taking the standard deviation function from Table 1 as γ. **a** Coherent user ($s(u_1) = C^{\sigma}_{u_1} = -1.28$) and **b** Not coherent user ($s(u_2) = C^{\sigma}_{u_2} = -5.95$)

The magic barrier of the system (and the corresponding magic barrier of each user) was estimated according to the procedure described in Algorithm 1. For this, the ratings and opinions of those user-item pairs for the recorded 6299 opinions were used. This setup corresponds to sampling two ratings for each of the 6299 user-item pairs, that is, $m = 2$ and r^1_{ui} corresponds to the rating provided by user u to item i in the system *before* the survey, whereas r^2_{ui} corresponds to the opinion collected during the survey. The estimate of the magic barrier is the average of the squared sample noise over all 6299 user-item pairs.

3 A measure of user coherence for recommendation

Given a user u, her rated items $I(u) \subseteq \mathcal{I}$, and the ratings $r(u, i)$ assigned to these items, i.e., $i \in I(u)$, we aim to provide a score $s(u) = s(I(u))$ that measures how coherent a user is, in terms of her assigned ratings. To compute such score we propose to use an external information source, upon which we measure the inconsistencies of the user's ratings, by describing items in terms of features such as genres. Furthermore, we aim to use only one item attribute. This is because in several domains—or, at least, in typical public datasets—there are usually no more than one or two features available for each item. Thus, by using only one feature, our model could be applied to as many scenarios as possible. We show in the rest of the paper that this measure obtains very good results, despite its seemingly simple nature. Nonetheless, it should be noted that a combination of features may lead to a better solution than using only one as we shall study here, but we leave this possibility as future work, as it is out of the scope of the present work.

3.1 Example

Before presenting the actual definition of the rating coherence of a user (or *user coherence*, for simplicity), let us first consider the examples presented in Fig. 1. Here we have two users, that have rated exactly the same set of items, although their ratings are slightly different. Specifically, the user in Fig. 1a exhibits a more consistent rating pattern to items sharing the same features, which in this case corresponds to the movie's genres. In the long term we argue that such a user will have a more consistent (less noisy) behavior, since her tastes for each item feature seem to be well defined.

3.2 User coherence

Following the rationale presented before, we define the user coherence based on a set of item features or attributes $\mathcal{A}$ as:

$$C_u^\gamma = -\sum_{a\in\mathcal{A}} \gamma_{u,a} \tag{7}$$

The user coherence C_u^γ, hence, provides a score that aims to measure how coherent a user profile u is in terms of her assigned ratings. It should be noted that we can use any arbitrary function γ defined upon the information known for a user u and a specific attribute a. This information will generally be the ratings given by u to the items associated with a, that is $u^a = \{r_{u,i}, a \in \mathcal{A}_i\}$, where $\mathcal{A}_i$ denotes the subset of attributes in $\mathcal{A}$ for item i. Further notation needed includes $u^{\mathcal{A}}$ denoting the user's ratings associated with any attribute in the space $\mathcal{A}$.

User coherence as measured by C_u^γ provides a generic formulation to incorporate any information available about the user—including external information such as item attributes—which is readily available in any recommender system. Although other measures could be available where no external information is required, such as the entropy of the ratings or the Kullback-Leibler divergence between the user's preferences and the overall preferences (Bellogín et al. 2011), we believe that our formulation provides a measure which is easily explainable and justifiable, allowing for further feedback from the recommender system to the user.

Table 1 shows some possibilities for these functions applied over the vector of ratings u^a, considering the formulation already presented and the probabilities $p(a|u)$ and $p(a)$, which are computed normalizing the rating values of a user or of the whole community for a given attribute. Entropy, Kullback-Leibler divergence (KLD), standard deviation, mean, and size are presented in the table, along with two weighted versions of the standard deviation and mean to consider for the actual number of items rated by the user in each attribute, hence, accounting for the importance of each feature in the user profile. To illustrate the impact of the weighted variations, we now show the computations of the coherence function C_u^γ for the two users in Fig. 1 when using the standard deviation or its weighted counterpart as functions γ:

$$\gamma = \sigma(u^a) \qquad \Longrightarrow \begin{cases} C_{u_1}^\gamma = -\left(\sigma(4,4,3) + \sigma(1,2) + \sigma(3,3)\right) = -1.28 \\ C_{u_2}^\gamma = -\left(\sigma(5,4,3) + \sigma(1,5) + \sigma(2,5)\right) = -5.95 \end{cases}$$

$$\gamma = \sigma(u^a)\frac{\|u^a\|}{\|u^{\mathcal{A}}\|} \qquad \Longrightarrow \begin{cases} C_{u_1}^\gamma = -\left(\sigma(4,4,3)\frac{3}{6} + \sigma(1,2)\frac{2}{6} + \sigma(3,3)\frac{2}{6}\right) = -0.52 \\ C_{u_2}^\gamma = -\left(\sigma(5,4,3)\frac{3}{6} + \sigma(1,5)\frac{2}{6} + \sigma(2,5)\frac{2}{6}\right) = -2.15 \end{cases}$$

Table 1 Possible functions $\gamma_{u,a}$ to be used in Eq. (7), following the notation described in the text

Function $\gamma_{u,a}$	Definition	Function $\gamma_{u,a}$	Definition
Entropy	$p(a\|u) \log p(a\|u)$	KLD	$p(a\|u) \log \frac{p(a\|u)}{p(a)}$
Mean	$\mu(u^a)$	Weighted mean	$\mu(u^a) \frac{\\|u^a\\|}{\\|u^{\mathcal{A}}\\|}$
SD	$\sigma(u^a)$	Weighted SD	$\sigma(u^a) \frac{\\|u^a\\|}{\\|u^{\mathcal{A}}\\|}$
Size	$\\|u^a\\|$		

We observe that, for both computations, coherence of user u_2 is smaller than that of user u_1.

The key aspect of the user coherence, hence, is that we are accounting for the rating deviation (or any other statistical measure on those ratings) of the user with respect to a particular feature space. Besides, such a definition allows for a more general case, where the space $\mathcal{A}$ could be—instead of (textual) item features—any embedding of the items into an space $\mathcal{A}$, such as an item clustering or the latent factors of the items.

We have to emphasize that any of these variations of coherence can be calculated using the same data available at training time by the recommender system, and that no information from the test set is needed or required.

3.3 Linking magic barrier and user coherence

There exists a direct connection between the definition of the magic barrier (Eq. (6)) and the proposed user coherence (Eq. 7): by taking the following function

$$\gamma_{u,a} = \sum_{(u,i)} - \left(\widehat{\mu}_{ui} - r^a_{ui}\right)^2, \tag{8}$$

we get $C^\gamma_u \approx \widehat{B}_u$, where the attribute space $\mathcal{A}$ corresponds to the timestamps when an item has been rated, that is, $\mathcal{A} = \{t : \exists i \; r^t_{ui} \in \mathcal{R}\}$, and $\widehat{B}_u$ is computed as in Eq. (6) but restricting $\mathcal{X}$ to those pairs where user u is present. This information is unknown in advance, and, in principle, it is not going to be similar to the item embedding we are using in this work (textual item attributes); nonetheless, it is relevant to note that a proper attribute space may give more accurate predictions, due to its mathematical equivalence.

4 Validating the user coherence

In this section, we assess the validity of the proposed coherence functions as good estimators for the magic barrier. For this, we use the dataset described in Sect. 2.3, where the magic barrier is available. As explained in that section, the magic barrier was estimated by performing a user study to collect users' *opinions* on items from the system. More specifically, we used the more than 10 million ratings available in

Table 2 Spearman's correlation between coherence and the user magic barrier

Coherence	Genres	Intended audience	Plot keywords
Entropy	0.050	0.048	0.000
KLD	0.098	0.067	0.068
Mean	0.114	0.097	0.106
Weighted mean	0.010	0.072	– 0.028
SD	– 0.331	– 0.383	– 0.279
Weighted SD	– **0.398**	– **0.432**	– **0.394**
Size	0.077	0.066	0.088
Random		– 0.015	
Number of ratings		– 0.072	
Average rating		– 0.104	

Best values (in absolute terms) are in bold

that dataset to compute the rating coherence (more statistics presented further on in Table 5), and compare those values against the magic barrier estimates for each user, by considering the 6299 opinions obtained through the user study.

In order to check how well our approach estimates the magic barrier, we used correlation coefficients. We show in Table 2 the Spearman's correlation values between the rating coherence and the magic barrier per user (Pearson's correlation values were similar).[9]

We observe in Table 2 that, among the different variations for the rating coherence, the correlations for the weighted version of the coherence function (that is, where the importance of each feature in the user profile is considered) show more predictive power only when the standard deviation is used. Besides, entropy and KLD do not perform very well, probably because the probability model defined by $p(a|u)$ is not powerful enough to capture the nuances present in the feature space; in the future we would like to test more complex models, e.g., based on Relevance Modeling (Parapar et al. 2013) or linked to performance prediction (Bellogín et al. 2011). Additionally, the *Intended Audience* feature seems to be the best feature space for some of the coherence formulations, and especially, for the cases where a strong correlation is obtained. We have to note, however, that this feature space offers a low coverage in terms of the items identified with this feature (Said et al. 2011), thus this aspect should also be taken into account. In fact, other features available in the system such as *Emotion keywords* or *Time keywords* are not included in this analysis because of their very low coverage.

We have also included in the comparison a random magic barrier predictor to check its neutral correlations (around zero), along with two other baseline predictors based on the number of ratings each user has and her average rating. These results show that

[9] Note that Pearson's correlation coefficient is designed to capture linear relationships between the two variables whereas Spearman's captures non-linear dependencies. Both correlation measures provide scores in the range of – 1 to 1, where 1 denotes a perfect correlation, – 1 represents an inverse correlation, and the absolute value is the magnitude, or strength, of the relationship.

the proposed coherence function is not trivial, and it is actually capturing something that other transformations based on the same information (mainly ratings) are not able to provide.

This experiment hence confirms that the proposed user coherence provides good predictions of the magic barrier; as a consequence, we should be able to exploit the ranking generated by sorting the users according to their coherence value to lower the magic barrier for the more coherent (or easy) users. In the next section, we show that this may be generalized when no information about the magic barrier is available, and only the final RMSE of the system can be measured.

5 Exploiting user coherence to improve recommendation performance

In this section we exploit the user coherence as a signal to separate users into different partitions. We assess the benefits of such partitions by creating several training and test models according to these clusters (Sect. 5.1) and analyze if the performance of a recommender system is improved depending on how the rating information from each of the clusters is combined (Sect. 5.3). Section 5.2 presents the experimental setup used in this analysis.

5.1 Coherence-based user clustering and data splitting

It was observed in Sect. 4 that user coherence obtains strong correlations with the magic barrier. However, that type of validation is limited, since it requires having a measure of user noise, which generally is not available. Hence, we now propose to validate the user coherence by measuring the change in performance of a recommender system where users are included in training/test splits according to their coherence values. By doing so, we can confirm the usefulness of our metric even in datasets where no measure of user noise is available.

With this goal in mind, we separate the users into either those *easy* to recommend to or *difficult* to recommend to. We do this by considering the ranking obtained using their coherence score and taking a percentage of them as easy (those with higher score) and difficult (those with lower score). More specifically, for any given percentage p, $p \cdot |\mathcal{U}|$ of the users are labeled as difficult, and the rest—i.e., $(1 - p) \cdot |\mathcal{U}|$—as easy. Then, we build four sets of ratings as in Fig. 2, two corresponding to the training and test splits for the easy users (Tr_e and Te_e) and two corresponding to the difficult users (Tr_d and Te_d).

Once these four rating partitions are generated, we build five combinations for training and test models as presented in Table 3. The rationale of these models goes as follows: the *All–All* model simulates the standard evaluation split where all the users are used to train and test a specific recommender system; that is, the user clustering does not affect the users being evaluated—although it may affect the number of ratings available for training or test, as we shall show later. *All–Easy* and *All–Diff* consider the same training set—more specifically, the same training used by the *All–All* model—and only the test set changes: whereas *All–Easy* only evaluates on easy users, *All–Diff* does the same but for difficult users. These models allow us to empirically check if

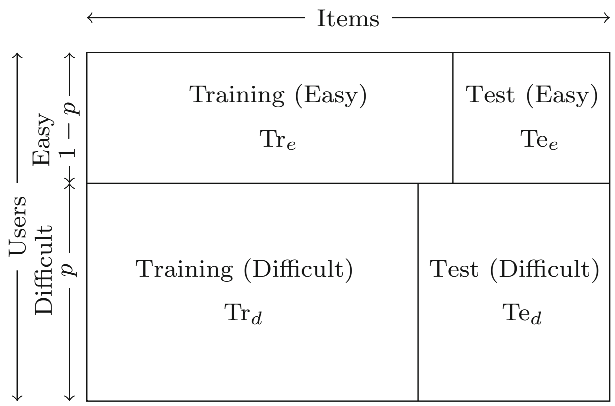

Fig. 2 How the data set is split into training and test sets. p denotes the percentage of users classified as difficult

Table 3 Notation for the different training and test models considered, where the name of each model represents [users in training set]–[users in test set]

Name	Training	Test
All–All	$\mathrm{Tr}_e \cup \mathrm{Tr}_d$	$\mathrm{Te}_e \cup \mathrm{Te}_d$
All–Easy	$\mathrm{Tr}_e \cup \mathrm{Tr}_d$	Te_e
All–Diff	$\mathrm{Tr}_e \cup \mathrm{Tr}_d$	Te_d
Easy–Easy	Tr_e	Te_e
Diff–Diff	Tr_d	Te_d

the clustering based on coherence values is actually separating the users into easy or difficult users, since the training set used by the recommender is the same.

Finally, the *Easy–Easy* and *Diff–Diff* evaluation models focus only on one user type, by training and testing on the ratings associated with the corresponding type of users. Actually, we may try to evaluate these models on the full test split, however, since no ratings exist in the training set for the other type of users, most recommender systems—i.e., collaborative filtering algorithms—will not be able to produce recommendations for those users, which means that we can directly ignore them and propose a test set that only contains users from the same type as those in the training set.

Any standard evaluation methodology can be used on top of these evaluation models. In this paper, we compute the root-mean-squared error (RMSE) of the recommendations produced for the test model; additionally, to reduce the variability of the results, we perform a fivefold cross validation on the set of ratings corresponding to each user. Other evaluation metrics and randomization or rating partition approaches can be used instead of the ones used here—such as time-aware evaluation strategies (Campos et al. 2014), this is however not in the scope of this work.

5.2 Experimental setup

We now empirically compare the RMSE of three standard collaborative filtering methods under the different evaluation models presented in Table 3. We use an item-based

Table 4 Recommendation algorithms and parameters used in the experiments

Algorithm	Short name	Parameter	Value
User-based nearest neighbor	UB	Neighborhood size	50
Item-based nearest neighbor	IB	Neighborhood size	50
Matrix factorization	MF	Factors	50

Table 5 Statistics on the datasets used in the experiments

Dataset	Movielens	Moviepilot	Yelp
Number of users	6040	318,418	45,981
Number of items	3900	31,948	11,537
Number of ratings	1,000,209	12,825,203	229,907
Density	4.24%	0.13%	0.04%
Range of ratings	[1–5]	[0–100]	[1–5]

nearest-neighbor recommender (Desrosiers and Karypis 2011) or IB, a matrix factorization method (Koren and Bell 2011) or MF, and a user-based nearest-neighbor recommender (Desrosiers and Karypis 2011) or UB. The first two are well-known for their efficiency and do not rely on the user dimension directly, which may provide further evidence that coherence is a useful criterion to classify users, an overview of the algorithms and parameters is shown in Table 4. Since we test different datasets we will not optimize their parameters and use the typical ones from the literature: Pearson similarity for IB and UB, 50 neighbors for UB, and 50 factors for MF. Because of this, no validation split is used to optimize any of these parameters, even though no tested method actually looks into the test split.

For the experiments, we have used the three datasets described in Table 5: Movielens, Moviepilot, and Yelp. The first is one of the datasets provided by Movielens[10], containing one million ratings, more than 6000 users and almost 4000 items. The second dataset, Moviepilot, was already described in Sect. 2.3, although we have to note that in these experiments the *opinions* gathered through the user study will not be considered. Additionally, we have to note that Moviepilot applies a linear transformation from its 0–10 rating scale with step size 0.5 (as described in Sect. 2.3) to a 0–100 rating scale with step size 5 (as seen in Table 5) in order to store ratings as integer values. In our experiments, we use the 0–100 values. The third dataset, Yelp[11], contains user reviews of places of business along with almost 230,000 ratings by more than 45,000 users on circa 11,000 points of interest (restaurants, cafés, shops, etc.).

In this experiment, the weighted standard deviation formulation will be used for the coherence function presented in (7) since it obtained stronger correlations with respect to the magic barrier (see Sect. 4). It might perhaps be interesting to compare other, simpler, baselines for this user-based clustering in combination with other variations

[10] http://www.grouplens.org/node/73.

[11] http://www.kaggle.com/c/yelp-recsys-2013/data.

of rating coherence. However, we focus on the evidence obtained before that the user coherence models achieved high correlation values with respect to the magic barrier. We particularly focus on the rating coherence based on weighted standard deviation, which obtained the strongest correlation against the magic barrier, hence, it would be the best candidate to use as a surrogate of the magic barrier. We plan to test, in the future, how other baselines or rating coherence functions, with lower correlations, perform in this type of experiment.

To compute the rating coherence (weighted standard deviation), we exploit dataset-specific features in each of the three datasets: for Movielens, we use the movie genres; for Moviepilot, we use three of the tagging features available in the dataset with more item coverage [genres, plot keywords, and intended audience (Said et al. 2013)]; finally, for Yelp we use the business categories, e.g., airports, tobacco shops, hospitals, etc. We should note that the nature of these item features is very different in each of these datasets—for instance, business categories are not as specific as item genres or plot keywords—and hence they encode the items differently, producing different representations for the user coherence and, possibly, they may perform better or worse depending on the quality of these features.

5.3 Effect in performance of a user clustering based on rating coherence

In this section we address the research question **RQ2** (see Sect. 1), where we investigate if we can improve the recommendation performance by clustering the population of users according to their coherence. We do so by, first, performing a thorough analysis on the Moviepilot dataset where different feature spaces are compared using an item-based recommender system (Sect. 5.3.1); and then, by testing three collaborative filtering algorithms on the Movielens and Yelp datasets, where we analyze the extent to which rating coherence affects CF algorithms in a different way (Sect. 5.3.2).

It is worth noting that the three datasets have different characteristics that prevent performing exactly the same experiment in all of them; more specifically, the item features are different: the only two that might be comparable are genres in Movielens and Moviepilot. Hence, since we cannot provide a cross-dataset comparison, we prefer to exploit a different dimension in each dataset: in Moviepilot we test the features that were previously compared against the magic barrier in terms of correlations (Sect. 4), in Movielens we test different algorithms because it is widely used and these results could be contextualized in the literature, and, finally, in Yelp only one of the methods (MF) is tested because all the alternatives have been previously analyzed, and this is one of the most well-known and best performing methods in the area (Koren and Bell 2011). These three, alternative but compatible experimental conditions do show that our method can be applied to different domains and the results are, to a large extent, positive and consistent.

5.3.1 Sensitivity to the feature space

Figure 3 shows the performance of the five training and test models introduced in Sect. 5.1 where three different attribute spaces are used in Moviepilot with an item-

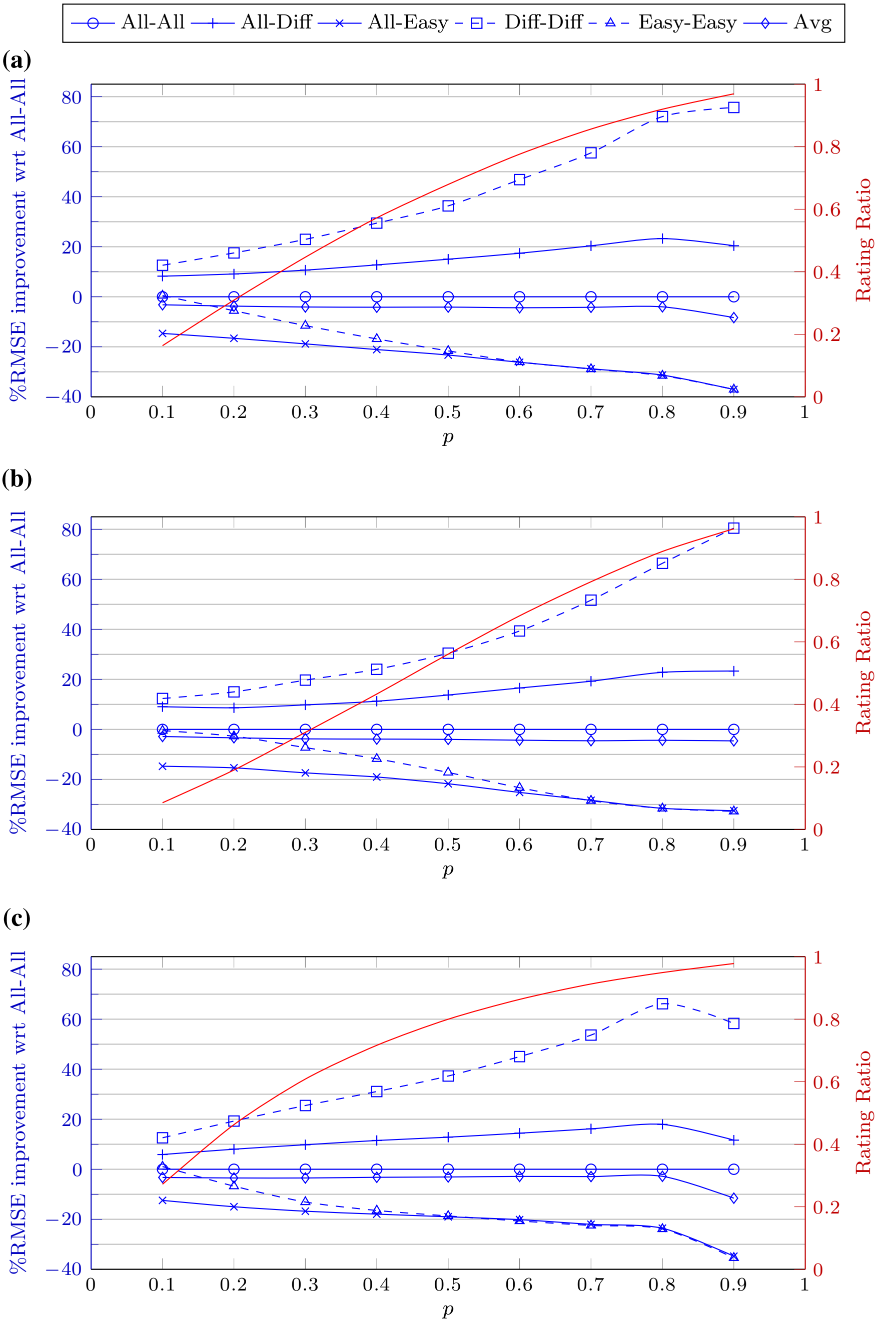

Fig. 3 Comparison of RMSE improvement with respect to the All–All split model for an item-based CF method, when different numbers of users are considered as difficult (p) and three attribute spaces are considered (genres, intended audience, and plot keywords) in the Moviepilot dataset. **a** Genres, **b** intended audience and **c** plot keywords

based collaborative filtering algorithm (IB). Figure 3a shows the results for Genres, Fig. 3b for the Intended Audience space, and Fig. 3c for Plot Keywords. We also include the number of ratings corresponding to each percentage of difficult users p and named it *rating ratio* (represented as a solid line); it is computed as the ratio of the number of ratings assigned to difficult users in a particular split, that is: $|\mathrm{Tr}_d|/|\mathrm{Tr}_e \cup \mathrm{Tr}_d \cup \mathrm{Te}_d \cup \mathrm{Te}_e|$.

As an additional visual clue, we included the weighted average error between *All–Easy* and *All–Diff* (line denoted as *Avg*), computed considering the number of users (p or $(1 - p)$) each model contains in its test set, in such a way that the resulting error should be comparable to the one obtained for *All–All*. This value aims to capture a weighted average that weights based on the percentage p, and thus, it would consider each test user with the same weight in the final computation, leading to a comparable error as *All–All*, since in that case every user appears at least once in the test set.

Note that, as explained before, the *All–All* split would simulate a scenario where the user clustering does not affect the users being evaluated, however, since the user clustering is always used, there might be some fluctuation in the error obtained in each case; in particular, the error obtained for the *All–All* split does not necessarily remain constant for any value of p, and may depend (and change) on the algorithm and the dataset. Because of this, we should use the error value of the *All–All* model at each percentage point as the base error to compare against. To further make this comparison easier to understand, from now on we report the relative error of other training and test models with respect to this split, which means that, if the error for *Diff–Diff* is 35.2 for $p = 0.3$ using Genres, we report a relative error of $(35.2 - 28.6)/28.6 = 23.08\%$, since the base error of *All–All* is 28.6.

In these figures, we observe that the general trend remains the same in any attribute space: more coherent users tend to have a better performance than the overall system (*All–Easy* has always a negative improvement of RMSE with respect to *All–All*, meaning that its RMSE is lower), which in turn performs better than the less coherent users (*All–All* lower than *All–Diff*). We should also note that, even though the strongest correlation between the coherence function was obtained for the Intended Audience attribute space, it seems the best performance now is found when Genres are used. However, the differences in performance between these two attribute spaces is not very large (the best RMSE is, in both situations, for *All–Easy* with $p = 0.9$, with a value of 11.8 using Genres and 12.5 using Intended Audience), which matches their correlations being too similar to each other (-0.398 vs. -0.432).

On the other hand, an aspect that is different in each attribute space is the rating ratio. The figures show that, depending on the attribute, for a fixed number of users, say $p = 0.5$, the number of ratings corresponding to those users might be very different. For example, around 68% of the ratings correspond to the difficult users using Genres, whereas this number goes down to 56% using Intended Audience or up to 80% using Plot Keywords. This indeed affects how steep the lines are in each model, since this is related to the number of ratings in training at each point, and hence, to the amount of information available by the recommendation algorithm. Therefore, the feature space plays an important role in the coverage of the recommendation algorithm, although it does not affect their performance much (see above), it could produce splits with very

Table 6 Improvement on rating prediction performance on Movielens using Genres as attribute space, for the different training and test models described in Sect. 5.1, where the ratio of difficult users is $p \in \{0.1, 0.5, 0.9\}$ and the base error column denotes the error obtained by the All–All model

Rec	p	Base error	All–Diff	All–Easy	Diff–Diff	Easy–Easy	Avg	Rating ratio
IB	0.1	0.99	31.41	– 3.84▲	70.40▼	– 2.83	13.84	0.09
IB	0.5	0.99	10.60	– 15.24▲	14.13▼	– 11.60	– 2.32	0.54
IB	0.9	0.99	1.41	– 31.58▲	2.02▼	– 19.88	– 15.04	0.94
UB	0.1	1.08	26.16	– 3.05	29.85▼	– 4.25▲	11.55	0.09
UB	0.5	1.09	9.82	– 10.55	12.48▼	– 14.40▲	– 0.37	0.54
UB	0.9	1.08	1.38	– 17.34	2.21▼	– 25.18▲	– 8.03	0.94
MF	0.1	0.89	31.54	– 4.04▲	54.32▼	– 3.82	13.69	0.09
MF	0.5	0.89	9.53	– 14.57▲	13.90▼	– 12.11	– 2.47	0.54
MF	0.9	0.89	1.57	– 28.51▲	1.91▼	– 22.67	– 13.47	0.94

▲ and ▼ denote the best and worst values obtained for each configuration (the lower the error, the better) respectively

little information for a percentage of the users. The following section highlights that this could be an issue in some datasets.

5.3.2 Analysis of different collaborative filtering algorithms

In this section we analyze the effect of user coherence under different recommendation algorithms. For this, we use first the Movielens dataset, and later the Yelp dataset. Both datasets do not contain information regarding the magic barrier of the users, but we are able to apply the same methodology as in the previous section (by computing how performance changes when a recommender is applied to different splits of the same data) to validate our approach.

As before, we report the relative RMSE values for the different training and test models presented in Table 3 with respect to the *All–All* model. Table 6 shows three sets of values for each algorithm being tested in Movielens: one when the 10% least coherent users are classified as difficult users ($p = 0.1$), another when the top 10% coherent users are classified as easy users (thus, the remaining 90% are classified as difficult users, or $p = 0.9$), and one where half of the users are classified as difficult and the remaining as easy ($p = 0.5$).

In this context, we observe that more coherent (easier) users tend to have a better performance than the overall system (note that *All–Easy* improvements are always negative in Table 6, meaning that the error is always lower than *All–All*) for every recommender. For the same reason, the *Diff–Diff* model is always the worst performing one. This is because the users belonging to this model are the noisiest ones, and when the recommender only uses this type of ratings, it would produce worse recommendations than when information from the rest of the community is considered (compare the error values between *All–Diff* and *Diff–Diff*).

We also observe that difficult users need additional information—i.e., not only from other difficult users—to reduce their error (see *Diff–Diff* vs. *All–Diff*), whereas easy

Table 7 Improvement on rating prediction performance on Yelp using categories as attribute space and MF as recommendation algorithm, using notation from Table 6

p	Base error	All–Diff	All–Easy	Diff–Diff	Easy–Easy	Avg.	Rating ratio
0.1	1.26	8.60▼	− 8.44▲	7.64	− 7.96	0.08	0.40
0.5	1.26	0.32▼	0.00	− 0.71▲	0.08	0.16	0.87
0.9	1.26	0.16▼	− 2.15	− 0.08	− 4.70▲	− 1.04	0.97

users appear to be more stable (*Easy–Easy* vs. *All–Easy*). This is clearer for $p = 0.5$, since the same number of users is used in both groups, and thus, the computed errors are comparable.

The difference in performance is even more evident when we compare *All–Easy* versus *All–Diff* with $p = 0.5$, where there always is a performance improvement; taking into account that the number of users is the same (which in this case also correspond to very similar sizes of the test sets, see the *Rating Ratio* column), these results show that, even when the recommender has the same information to train with, its actual error only depends on the type of users we use to evaluate it. In this case, more coherent (easy) users obtain more accurate recommendations.

It is important to emphasize that the effect of coherence-based user clustering is evident in the three collaborative-filtering algorithms tested. Whereas UB depends inherently on the users in the system by the similarity being used and the neighborhood involved during the recommendation phase, both IB and MF are well-known to not rely directly on other users (besides the target user) to generate a recommendation. Hence, this shows that the proposed formulation is general enough to work on different types of recommendation algorithms.

To further evidence the generality of our proposal, we include results for the Yelp dataset in Table 7. Here we focus on the MF algorithm for a number of reasons. First, because as presented in Table 6, this algorithm is the best performing one, and according to the literature, it tends to perform very well in many situations involving rating-based datasets and comparing against other collaborative filtering algorithms (Koren and Bell 2011). Second, and more importantly in this situation, it is the only one that shows full recommendation coverage in all the reported situations. In an exploratory study, we observed that UB and IB were only able to produce recommendations for Yelp in certain situations, i.e., for particular values of p. However, MF, has full coverage by definition, since it is able to work as soon as a user or item has one rating in the system.

This is a critical issue in this dataset, because the rating ratio is highly skewed (the reader should note that it is already up to 87% for $p = 0.5$) which produces splits with very few ratings when fewer and fewer users are being considered as easy (larger p). One possible reason for this behavior is that the only available item attribute in this dataset does not have the same inherent meaning than the other attributes used in the previous datasets; more specifically, in a movie or music domain, an attribute such as genre allows to classify the same type of items into different categories, however in Yelp we actually have very different types of items (hotels, restaurants, museums, etc.) which end up, in this dataset, receiving ratings in a highly skewed, biased way for both users and items.

Because of this, the results found in this dataset are slightly different to those reported before. More specifically, when $p = 0.5$ the best performing model is *Diff–Diff*. Note that, even in this situation, the performance of *All–Diff* is much worse, and *All–All* and the models involving easier users (*All–Easy* and *Easy–Easy*) have a very similar performance, something that never occurred in the previous examples, evidencing this atypical scenario produced by a higher rating density for the difficult users. This highly skewed distribution, as evidenced by the large amount of ratings assigned to difficult users, implies that the easy users considered in the *All–All* and *All–Diff* splits do not contribute with many ratings, and hence, they are not able to compensate the incoherent behavior learnt by the model from the difficult users.

For the other cases presented in Table 7 ($p = 0.1$ or 0.9), either *All–Easy* or *Easy–Easy* models perform the best. This is in line with the results presented before for Movielens and Moviepilot, even though the rating distribution in these three datasets is not comparable. We hence conclude that the proposed user clustering based on rating coherence is also useful in the Yelp dataset, however more attention should be paid to the rating distribution and the nature of the item attributes, since these two factors affect the extent of the improvements of the tested models. For instance, the largest improvement was obtained for *All–Easy* with $p = 0.1$ (the RMSE was decreased by 8%), however, in Movielens up to 30% improvement can be achieved by different recommenders, a similar situation to the one obtained in Moviepilot.

5.4 Discussion

In summary, this experiment confirms that it is possible, in most cases, to exploit the coherence values to build different training (and test) models in such a way that the error decreases for the easy users, i.e., to increase the accuracy of the recommender system. In particular, we have observed that it is possible to improve the performance for 90% of the user population simply by creating a separate training set for the 10% noisiest (least coherent) users in Movielens and Yelp, so that the rest of the users can benefit from more consistent ratings (*Easy–Easy* model with $p = 0.1$). We have also observed that, for the 10% most coherent users, if trained separately, we will always obtain an improvement in performance. This would reduce the amount of resources needed for these users and could be used instead to finely tune the algorithms for the less coherent part of the user population.

Additionally, it should be noted that no modification was made to any of the algorithms reported, and that is why the proposed clustering—and specifically, the user coherence measure that allows such clustering—is presented as a good alternative to discriminate between easy and difficult users. Similar improvements can be expected from other recommendation techniques, and even larger improvements if specially tailored algorithms are introduced to deal with the noisier characteristics of the more difficult users.

Furthermore, in contrast with other techniques presented in the past [such as significance weighting or trust-based recommendation (Herlocker et al. 2002; O'Donovan and Smyth 2005)], the proposed technique (rating coherence) is able to discriminate between easy and difficult users according to, and independently from, different recommendation techniques. Effectively, this means that users with higher rating coherence (hence, labeled as easy) tend to receive better recommendations than their counterpart users with low rating coherence. This was shown for different datasets and three recommendation techniques (user-based and item-based nearest neighbors and matrix factorization) that use different information to base their recommendations.

6 Related work

The idea of predicting the performance of recommender systems has attracted a lot of attention in the field, as we show in the next sections, however, to the best of our knowledge, no other work has been able to predict an actual measure of users' inconsistency—like the magic barrier—and successfully apply it to improve the performance of the whole (or even of a subset) of the users in the system, as we have presented in this work.

6.1 Measuring user uncertainty through ratings

Inconsistency in user behavior in the context of information and recommender systems is a known concept and has been studied on several occasions previously. The first mention of inconsistencies in a scope similar to ours was made in (Hill et al. 1995) in their study on virtual communities. The authors questioned how reliable the ratings were and found a rough estimate by calculating the RMSE between two sets of ratings performed by 22 users on two occasions 6 weeks apart.

Similar reliability issues, e.g., the levels of noise in user ratings, were discussed in (Herlocker et al. 2004), coining the term *magic barrier* as an upper level of recommender system optimization. More recently, in (Amatriain et al. 2009a) the authors performed a set of user trials on 118 users based on a subset of the Netflix Prize dataset. They attempted to find answers to whether users are inconsistent in their rating behavior, how large the inconsistencies are, and what factors have an impact on the inconsistencies. They were able to identify a lower bound—the magic barrier—for the dataset used in the trials.

Following their user trials, in (Amatriain et al. 2009b) the authors successfully increased the accuracy of a recommender system by implementing a de-noising step based on re-ratings collected in a study. They presented two re-rating strategies (user-based and data-based) in order to find the *ground truth values* of ratings for the purpose of maximizing accuracy improvements in a recommender system. They concluded that re-rating previously rated items could, in some circumstances, be more beneficial than rating previously unrated items. In a different work, Said et al. (2012b) attempted to estimate the magic barrier of a system using a user study within a real-world recommendation system (as presented in Sect. 2.3). In (Jasberg and Sizov 2017), the authors extend these models and propose a probabilistic framework by using methods from metrology and physics.

Some of the inconsistencies in users' rating behavior can be mitigated by temporal aspects, as shown in (Lathia et al. 2010). This mitigation does however not compensate for all inconsistencies, which was shown by other authors by having different time spans between re-ratings (Amatriain et al. 2009b).

In (Kluver et al. 2012) an information theoretic framework is proposed for estimating the amount of correct information that is contained in ratings and how much information is made up of noise and general user inconsistencies. The authors conclude that even though fine-grained rating scales (i.e., 1–5 stars) contain more noise than coarser scale (thumbs up/thumbs down), they contain more predictive power than the coarse-grained alternative. Other works propose a framework for classification of users based on the quality of their ratings, and then transfer learnt rating prediction models between user groups with higher quality data to user groups with lower quality data (Yu et al. 2016). Yet another approach to handle natural noise in user ratings, presented in (Toledo et al. 2016), proposes an adaptable fuzzy profiling method for improving recommendation accuracy.

6.2 Other sources of user uncertainty

Additional sources besides ratings have been proposed to estimate the user uncertainty. In (Nguyen et al. 2013) the authors explore various interfaces that support users in their rating behavior in order to minimize potential noise caused by rating inconsistencies. They conclude that it is possible to increase the quality and consistency of data by utilizing supportive functionalities such as tags and examples of previously rated items.

More recently, works like (Toledo et al. 2015) focus on detecting and correcting the natural noise in user ratings by linking the noise to the users' personality traits. The approach in (Saia et al. 2016) instead attempts to rid the user profile of ratings that appear to have been given in an incoherent manner in order to more accurately reflect the user's true preference profile.

Jones et al. (2011) propose a model where instead of ratings, recommendations are created based on comparisons between items, i.e., users compare pairs of items and state which one they prefer. This information is then used instead of ratings. The authors report their model to be more stable over time than systems only based on ratings.

Finally, from a psychological point of view, the inconsistency a person might have towards a certain item (recommended or not) can be related to a concept known as *cognitive dissonance*, which refers to the notion of a person that simultaneously holds two or more contradictory beliefs about a certain action or item. This concept has been discussed in the scope of information systems previously, in, e.g., (Bajaj and Nidumolu 1998) where the authors attempt to build guidelines for information system production taking cognitive dissonance into consideration.

6.3 Linking recommender performance to user uncertainty

Apart from measuring the uncertainties, other authors have explored the problem of understanding how the recommenders fail for certain users, by attempting to characterize those users. In (Rashid et al. 2005), the authors propose a measure of the effect of a user in the recommendations received by an algorithm, named as influence. Their original definition is very expensive, since it measures the effect a user has over the rest via the predictions they receive, for which they need to compute predictions for items using a training model where the target user has been removed.

In (Ekstrand and Riedl 2012), the authors examine why some recommenders fail in the context of hybrid recommendation, with the goal of selecting better components to build more efficient ensembles. They found that recommenders fail for different users and items, and obtained specific user features—such as the user's rating count, the average rating, and their variance—that allow to predict the performance of an algorithm.

(Kille 2012) assigns a difficulty value reflecting the expected evaluation outcome of the user. The work proposes to measure this difficulty in terms of the diversity of the rating predictions and rankings when comparing the output of several recommender systems. Some diversity metrics from (Kuncheva and Whitaker 2003) are proposed, but they are not tested nor implemented on real world datasets.

By drawing from Information Retrieval related quantities, a family of performance predictors for users is presented in (Bellogín 2011; Bellogín et al. 2011). Correlations found between ranking-based metrics and such predictors are strong, and the authors propose to exploit them in at least two applications: dynamic neighborhood building and dynamic ensemble recommendation, where the weights for the neigbors or the recommenders would dynamically change depending on the predicted performance of each variable.

More recently, a similar approach was developed using a machine learning method based on decision trees. In (Griffith et al. 2012) the authors aim to predict the user's performance in terms of the user's average error by extracting user's rating information (such as the number of ratings, average rating, standard deviation, number of neighbors, average similarity, etc.). The correlations obtained are very strong (around 0.8) but no actual applications are proposed in the work.

7 Conclusions and future work

The research presented here aims to provide a deeper understanding of what user characteristics are related with the appropriateness and relevance of the recommender's suggestions for each user. We have observed that being statistically coherent—in terms of *rating deviation*—within an item's attribute space (e.g., genres) gives enough information to predict the user's inconsistency as measured by her magic barrier. This opens up the possibility for a (real world) recommender system to perform different actions on the users depending on their predicted inconsistencies, such as proactively asking some specific users (the ones predicted as most *difficult*) to rate more items—by means of preference elicitation or active learning techniques (Rubens et al. 2011)—or training separate models for the *easy* and *difficult* users.

Our work shows that a recommender system can be trained differently depending on users' inconsistencies predicted by their rating coherence, allowing, for instance, cheaper recommendation cycles (in terms of computational effort, time, and parameter tuning) for easier users—i.e., those with higher coherence. In our experiments we observed that the rating prediction performance can be improved by a factor between 10 and 40% (depending on the dataset and the amount of users labeled as difficult) when only easy users are considered for training and testing the model. The remaining users (those labeled as difficult) will receive worse recommendations *in general* if the same algorithm is used as for the easier users, because of their lack of rating coherence; hence, other strategies—such as eliciting more preferences from them or using more complex algorithms for these users—should be exploited. This work has studied and analyzed the non-trivial task of discriminating between easy and difficult users. Testing the hypotheses that this work has arrived at is left for the future, once a specific separation between the two types of users has been properly defined.

Further, this work consolidates the concept of rating coherence and its effect on collaborative filtering recommender systems. In particular, we have obtained strong correlations between different variations of rating coherence and the estimated magic barrier, evidencing a higher predictive power than other, simpler baselines. When utilized correctly, this effect can be used to improve the quality of rating prediction, and potentially lower the computational effort needed. It can be argued that measuring the instability of user ratings over time would lead to a situation closer to the real world than what we have presented here. However, it should be noted that, even though the temporal dimension of the data is very important in recommendation, no real inconsistency in terms of *instability of user ratings over time* can be found in any public rating dataset because every user-item interaction is only presented once in rating datasets, probably storing only the last rating the user decided to give to an item. Furthermore, in some datasets [such as the well-known and widely used Movielens dataset (Harper and Konstan 2016)] the timestamps associated to the ratings are meaningless or artificial, and hence, temporal splits or temporal models on those datasets are not useful to capture the time dimension.

Nonetheless, the main advantage of the method presented here is that it is general enough to be applied to simple, realistic datasets, where such information is not usually available. This allows for maintaining decent correlations with respect to the measured magic barrier and decreasing the error for the users predicted to be easier. In any case, it would be interesting and relevant for the field to understand the users' rating inconsistencies over time, for which specific datasets with such information should become available. Once more datasets with these characteristics can be analyzed, it will be worthwhile to investigate whether the proposed measure of user coherence matches such inconsistency in user preferences. Our definition likely is a near approximation of the actual user uncertainty or inconsistency, as evidenced in our previous work on measuring the magic barrier by forcing repeated ratings over time (Said et al. 2012a). Knowing the real relation would be useful when building more accurate recommendation systems.

Moreover, the foci of this and earlier works on the magic barrier have been strictly on rating prediction, given the recent developments in the recommender systems sphere, rating prediction does not play a role as significant now as it once did. One line of future work we are currently studying is the application of the magic barrier, and similar concepts, on ranking-based recommendation scenarios. More explicitly, we are interested in evaluation using ranking metrics, e.g., precision, to analyze if the user's inconsistencies are also reflected in, or if they affect, their ranking performance; similarly, other performance indicators like utility or user satisfaction could be explored in the future and their relation with coherence analyzed.

The validation of our weighted standard deviation measure has only been possible on the single data set (Moviepilot) for which repeated ratings are available. A more substantial evaluation of this and the other coherence measures must wait until additional data sets containing such re-rating information become available. However, as evidenced by our experiments, the weighted standard deviation measure, relative to item attributes, effectively discriminates between more and less predictable users.

Additional lines of potential future work include the application of the concept of the magic barrier on domains outside of recommender systems, e.g., whether query-driven information retrieval suffers from similar problems, in the context, for instance, of the objective relevance assessments obtained, considering that it has already been observed that no complete inter-assessor agreement is found in general (Voorhees 1998; Webber et al. 2012). Similarly, by drawing a *user-as-a-query* analogy, a query analysis could be performed to investigate if some queries are systematically more difficult than others simply because their relevance assessment is less coherent than other queries, in line with other studies on hard topics and related TREC tracks (Voorhees 2004).

Acknowledgements The authors would like to thank Brijnesh-Johannes Jain at Technische Universität Berlin for support during this work and the anonymous reviewers for their helpful and constructive comments. This research was in part supported by the Spanish Ministry of Economy, Industry and Competitiveness (TIN2016-80630-P).

References

Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**, 734–749 (2005)

Amatriain, X, Pujol, J.M., Oliver, N.: I like it... i like it not: evaluating user ratings noise in recommender systems. In: Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization: formerly UM and AH, Springer-Verlag, Berlin, Heidelberg, UMAP '09, pp. 247–258 (2009a). https://doi.org/10.1007/978-3-642-02247-0_24

Amatriain, X., Pujol, J.M., Tintarev, N., Oliver, N.: Rate it again: increasing recommendation accuracy by user re-rating. In: Proceedings of the Third ACM Conference on Recommender Systems, ACM, New York, NY, USA, RecSys '09, pp. 173–180 (2009b). https://doi.org/10.1145/1639714.1639744

Bajaj, A., Nidumolu, S.R.: A feedback model to understand information system usage. Inf. Manag. **33**(4), 213–224 (1998). https://doi.org/10.1016/S0378-7206(98)00026-3

Bellogín, A.: Predicting performance in recommender systems. In: Proceedings of the Fifth ACM Conference on Recommender Systems, ACM, New York, NY, USA, RecSys '11, pp. 371–374 (2011). https://doi.org/10.1145/2043932.2044009

Bellogín, A., Castells, P., Cantador, I.: Predicting the performance of recommender systems: an information theoretic approach. In: Amati, G., Crestani, F. (eds.) Advances in Information Retrieval Theory, Lecture Notes in Computer Science, vol. 6931, pp. 27–39. Springer, Berlin (2011). https://doi.org/10.1007/978-3-642-23318-0_5

Bellogín, A., Said, A., de Vries, A.P.: The magic barrier of recommender systems—no magic, just ratings. In: Dimitrova, V., Kuflik, T., Chin, D., Ricci, F., Dolog, P., Houben, G.J. (eds.) User Modeling, Adaptation, and Personalization, Lecture Notes in Computer Science, vol. 8538, pp. 25–36. Springer International Publishing, Berlin (2014). https://doi.org/10.1007/978-3-319-08786-3_3

Campos, P.G., Díez, F., Cantador, I.: Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. User Model User-Adapt Interact. **24**(1–2), 67–119 (2014). https://doi.org/10.1007/s11257-012-9136-x

Desrosiers, C., Karypis, G.: A comprehensive survey of neighborhood-based recommendation methods. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 145–186. Springer, Berlin (2011)

Ekstrand, M., Riedl, J.: When recommenders fail: predicting recommender failure for algorithm selection and combination. In: Proceedings of the Sixth ACM Conference on Recommender Systems, ACM, New York, NY, USA, RecSys ’12, pp. 233–236 (2012). https://doi.org/10.1145/2365952.2366002

Griffith, J., O’Riordan, C., Sorensen, H.: Investigations into user rating information and predictive accuracy in a collaborative filtering domain. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing, ACM, New York, NY, USA, SAC ’12, pp. 937–942 (2012). https://doi.org/10.1145/2245276.2245458

Harper, F.M., Konstan, J.A.: The movielens datasets: history and context. TiiS **5**(4), 19:1–19:19 (2016). https://doi.org/10.1145/2827872

Herlocker, J.L., Konstan, J.A., Riedl, J.: An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. Inf. Retr. **5**(4), 287–310 (2002). https://doi.org/10.1023/A:1020443909834

Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. **22**(1), 5–53 (2004). https://doi.org/10.1145/963770.963772

Hill, W., Stead, L., Rosenstein, M., Furnas, G.: Recommending and evaluating choices in a virtual community of use. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, CHI ’95, pp. 194–201 (1995). https://doi.org/10.1145/223904.223929

Jasberg, K., Sizov, S.: The magic barrier revisited: accessing natural limitations of recommender assessment. In: Cremonesi, P., Ricci, F., Berkovsky, S., Tuzhilin, A. (eds.) Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27–31, 2017, ACM, pp. 56–64 (2017). https://doi.org/10.1145/3109859.3109898

Jones, N., Brun, A., Boyer, A.: Comparisons instead of ratings: towards more stable preferences. In: Boissier, O., Benatallah, B., Papazoglou, M.P., Ras, Z.W., Hacid, M. (eds.) Proceedings of the 2011 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2011, Campus Scientifique de la Doua, Lyon, France, August 22–27, 2011, IEEE Computer Society, pp. 451–456 (2011). https://doi.org/10.1109/WI-IAT.2011.13

Kille, B.: (2012) Modeling difficulty in recommender systems. In: Proceedings of the ACM RecSys 2012 Workshop on Recommendation Utility Evaluation: Beyond RMSE, RUE, pp. 30–32

Kluver, D., Nguyen, T.T., Ekstrand, M.D., Sen, S., Riedl, J.: How many bits per rating? In: Cunningham, P., Hurley, N.J., Guy, I., Anand, S.S. (eds.) Sixth ACM Conference on Recommender Systems, RecSys ’12, Dublin, Ireland, September 9–13, 2012, ACM, pp. 99–106 (2012). https://doi.org/10.1145/2365952.2365974

Koren, Y., Bell, R.: Advances in collaborative filtering. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 145–186. Springer, Berlin (2011)

Kuncheva, L.I., Whitaker, C.J.: Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Mach. Learn. **51**(2), 181–207 (2003)

Lathia, N., Hailes, S., Capra, L., Amatriain, X.: Temporal diversity in recommender systems. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, USA, SIGIR ’10, pp. 210–217 (2010). https://doi.org/10.1145/1835449.1835486

Masthoff, J.: Group recommender systems: aggregation, satisfaction and group attributes. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 743–776. Springer, Berlin (2015). https://doi.org/10.1007/978-1-4899-7637-6_22

McLaughlin, M.R., Herlocker, J.L.: A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In: Sanderson, M., Järvelin, K., Allan, J., Bruza, P. (eds.) SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, July 25–29, 2004, ACM, pp. 329–336 (2004). https://doi.org/10.1145/1008992.1009050

McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: Olson, G.M., Jeffries, R. (eds.) Extended Abstracts Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, April 22–27, 2006, ACM, pp. 1097–1101 (2006). https://doi.org/10.1145/1125451.1125659

Nguyen, T.T., Kluver, D., Wang, T., Hui, P., Ekstrand, M.D., Willemsen, M.C., Riedl, J.: Rating support interfaces to improve user experience and recommender accuracy. In: Yang, Q., King, I., Li, Q., Pu, P., Karypis, G. (eds.) Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12–16, 2013, ACM, pp. 149–156 (2013). https://doi.org/10.1145/2507157.2507188

O'Donovan, J., Smyth, B.: Trust in recommender systems. In: Amant, R.S., Riedl, J., Jameson, A. (eds.) Proceedings of the 10th International Conference on Intelligent User Interfaces, IUI 2005, San Diego, California, USA, January 10–13, 2005, ACM, pp. 167–174 (2005). https://doi.org/10.1145/1040830.1040870

Parapar, J., Bellogín, A., Castells, P., Barreiro, A.: Relevance-based language modelling for recommender systems. Inf. Process. Manag. **49**(4), 966–980 (2013). https://doi.org/10.1016/j.ipm.2013.03.001

Rashid, A.M., Karypis, G., Riedl, J.: Influence in ratings-based recommender systems: An algorithm-independent approach. In: Kargupta, H., Srivastava, J., Kamath, C., Goodman, A. (eds.) Proceedings of the 2005 SIAM International Conference on Data Mining, SDM 2005, Newport Beach, CA, USA, April 21–23, 2005, SIAM, pp. 556–560 (2005). https://doi.org/10.1137/1.9781611972757.60

Ricci, F., Rokach, L., Shapira, B.: Introduction to recommender systems handbook. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 1–35. Springer, Berlin (2011)

Rubens, N., Kaplan, D., Sugiyama, M.: Active learning in recommender systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 735–767. Springer, Berlin (2011). https://doi.org/10.1007/978-0-387-85820-3_23

Saia, R., Boratto, L., Carta, S.: A semantic approach to remove incoherent items from a user profile and improve the accuracy of a recommender system. J. Intell. Inf. Syst. **47**(1), 111–134 (2016). https://doi.org/10.1007/s10844-016-0406-7

Said, A., Kille, B., De Luca, E.W., Albayrak, S.: Personalizing tags: a folksonomy-like approach for recommending movies. In: HetRec '11, New York, NY, USA, RecSys, pp. 53–56 (2011)

Said, A., Jain, B., Narr, S., Plumbaum, T.: Users and noise: the magic barrier of recommender systems. In: Masthoff, J., Mobasher, B., Desmarais, M., Nkambou, R. (eds.) User Modeling, Adaptation, and Personalization, Lecture Notes in Computer Science, vol. 7379, pp. 237–248. Springer, Berlin (2012a). https://doi.org/10.1007/978-3-642-31454-4_20

Said, A., Jain, B.J., Narr, S., Plumbaum, T., Albayrak, S., Scheel, C.: Estimating the magic barrier of recommender systems: a user study. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, USA, SIGIR '12, pp. 1061–1062 (2012b). https://doi.org/10.1145/2348283.2348469

Said, A., Berkovsky, S., De Luca, E.W.: Introduction to special section on camra2010: movie recommendation in context. ACM Trans. Intell. Syst. Technol. **4**(1), 13:1–13:9 (2013). https://doi.org/10.1145/2414425.2414438

Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 257–297. Springer, Berlin (2011)

Toledo, R.Y., Mota, Y.C., Martínez-López, L.: Correcting noisy ratings in collaborative recommender systems. Knowl. Based Syst. **76**, 96–108 (2015). https://doi.org/10.1016/j.knosys.2014.12.011

Toledo, R.Y., Castro, J., Martínez-López, L.: A fuzzy model for managing natural noise in recommender systems. Appl. Soft Comput. **40**, 187–198 (2016). https://doi.org/10.1016/j.asoc.2015.10.060

Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)

Voorhees, E.M.: Variations in relevance judgments and the measurement of retrieval effectiveness. In: Croft, W.B., Moffat, A., van Rijsbergen, C.J., Wilkinson, R., Zobel, J. (eds.) SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24–28 1998, Melbourne, Australia, ACM, pp. 315–323 (1998). https://doi.org/10.1145/290941.291017

Voorhees, E.M.: Overview of the TREC 2004 robust track. In: Voorhees, E.M., Buckland, L.P. (eds.) Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004, Gaithersburg, Maryland, USA, November 16–19, 2004, National Institute of Standards and Technology (NIST), vol Special Publication, pp. 500–261 (2004). http://trec.nist.gov/pubs/trec13/papers/ROBUST.OVERVIEW.pdf

Webber, W., Chandar, P., Carterette, B.: Alternative assessor disagreement and retrieval depth. In: Chen, X., Lebanon, G., Wang, H., Zaki, M.J. (eds.) 21st ACM International Conference on Information and

Knowledge Management, CIKM’12, Maui, HI, USA, October 29–November 02, 2012, ACM, pp. 125–134 (2012). https://doi.org/10.1145/2396761.2396781

Yu, P., Lin, L., Yao, Y.: A novel framework to process the quantity and quality of user behavior data in recommender systems. In: Cui, B., Zhang, N., Xu, J., Lian, X., Liu, D. (eds.) Web-Age Information Management—17th International Conference, WAIM 2016, Nanchang, China, June 3–5, 2016, Proceedings, Part I, Springer, Lecture Notes in Computer Science, vol 9658, pp. 231–243 (2016). https://doi.org/10.1007/978-3-319-39937-9_18

Alan Said is Lecturer at The University of Skövde and a member of the university’s Artificial Intelligence research group. He holds a Ph.D. from Technishe Universität Berlin. His research spans the fields of user modeling, personalization, recommender systems, evaluation, and reproducibility. He has authored over 70 scientific publications in related fields.

Alejandro Bellogín is Assistant Professor at Universidad Autónoma de Madrid. Previously, he worked with Prof. Arjen de Vries associated to the Centrum Wiskunde & Informatica under a post-doctoral Marie Curie research grant. In 2012, he completed his Ph.D. under the supervision of Prof. Pablo Castells and Dr. Iván Cantador at Universidad Autónoma de Madrid. Dr. Bellogín has worked in several areas of user modeling and personalization, including recommender systems, evaluation, and reproducibility. His contribution in this work is based on experiences gained both from his Ph.D. work as well as his joint research with Dr. Said.